**----- class LentBookView-----**

/\*\*\* Submit registration information to controller  
\*/  
public static void submitRegistrationInfo()  
{

Submit registration information to controller

}

/\*\*\* Submit new borrow information to controller via POST request  
\*/  
public static void postNewBorrowInformation()  
{

Submit new borrow information to controller via POST request

}

/\*\*  
\* Notifies registration status to user  
\* @param message a string representation of the message to be notified to the user.  
\*/  
public void notifyRegistrationStatus(String message)  
{

Notifies registration status to user

}

/\*\*  
\* Notifies lent status  
\* @param message a string representation of the message to be notified to the user.  
\*/  
public void notifyLentStatus(String message)  
{

notify lent status to user;

}

**----- class BorrowBookController -----**

**/\*\***

\* Show the form for creating a new resource.

\*

\* @param Request $request

\* @return \Illuminate\Http\Response

\*/

public function create(Request $request)

{

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

$borrowRecord = (new BorrowRecord())→getBorrowRecordByID($request- >input('borrow\_record\_id'));

$borrowInformation = $borrowRecord->getAttribute('borrowInformation');

$currentDate = new DateTime();

$expiredDate = (new DateTime())->modify('+14 day');

if (is\_null($borrowInformation)) {

$newBorrowInformation = new BorrowInformation();

$newBorrowInformation->updateNewBorrowInformation(

$borrowRecord->getAttribute('id'),

$borrowRecord→getAttribute('user')→activeBorrowCard()- >getAttribute('id'),

$request->input('status\_before'),

$currentDate,

$expiredDate,

$request->input('borrow\_fee'),

$request->input('pre\_paid')

);

} else {

$borrowInformation->updateNewBorrowInformation(

$borrowRecord->getAttribute('id'),

$borrowRecord→getAttribute('user')→activeBorrowCard()- >getAttribute('id'),

$request->input('status\_before'),

$currentDate,

$expiredDate,

$request->input('borrow\_fee'),

$request->input('pre\_paid')

);

}

$borrowRecord->updateBorrowRecordStatus('lent');

$borrowRecord->getAttribute('bookCopy')->updateBookCopyStatus('lent', $request->input('status\_before'));

Session::flash('success', 'Lent book successfully!');

return redirect()->back();

}

**----- Class User -----**

**/\*\***

\* Get active borrow card

\* @return \Illuminate\Database\Eloquent\Model|null|static

\*/

public function getActiveBorrowCard()

{

return $this->borrowCards()->where('expired\_date', '>', new DateTime())

->first();

}

/\*\*

\* Get user with the user name

\* @param username of an user

\* @return \App\User

\*/

public function getUserByUsername(string $username)

{

return $this->where('name', $username)->first();

}

**----- Class BookCopy -----**

/\*\*

\* update the status of a book copy (available, referenced, borrowed or lent) and the status before lent in the database

\* @param copystatus status of a copy (available, referenced, borrowed or lent)

\* @param integrityStatus status before lent of a copy

\*/

public function updateBookCopyStatus(string $copystatus, string $integrityStatus = null)

{

$this->setAttribute('integrity\_status', $integrityStatus)

->setAttribute('copy\_status', $copystatus);

$this->save();

}

**----- Class BorrowRecord -----**

/\*\*

\* get the borrow record with status registered, match the id of the user and exist in the array of book copy ids in the database

\* @param userID id of an user

\* @param bookCopyIds array of book copy ids to search in

\* @return App\BorrowRecord match the query constains

\*/

public function getRegisteredBorrowRecord(int $userID, $bookCopyIds)

{

return $this->where('status', 'registered')

->where('user\_id', $userID)

->whereIn('book\_copy\_id', $bookCopyIds)->first();

}

/\*\*

\* search the borrow record by id in the database

\* @param borrowRecordID id of a borrow record

\* @return App\BorrowRecord

\*/

public function getBorrowRecordByID(int $borrowRecordID)

{

return $this->find($borrowRecordID);

}

/\*\*

\* update the status of borrow card to the database

\* @param borrowRecordID id of a borrow record

\* @return App\BorrowRecord

\*/

public function updateBorrowRecordStatus(string $status)

{

$this->setAttribute('status', $status);

$this->save();

}

**----- Class BorrowInformation -----**

/\*\* Update the new borrow information to the database

\* @param fields of borrow information object

\*/

public function updateNewBorrowInformation(int $borrowRecordID, int $borrowCardID, string $statusBefore, DateTime $lentDate, DateTime $expiredDate, int $borrowFee = null, int $prePaid = null)

{

$this->setAttribute('borrow\_record\_id', $borrowRecordID)

->setAttribute('borrow\_card\_id', $borrowCardID)

->setAttribute('status\_before', $statusBefore)

->setAttribute('lent\_date', $lentDate)

->setAttribute('expired\_date', $expiredDate)

->setAttribute('borrow\_fee', $borrowFee)

->setAttribute('pre\_paid', $prePaid);

$this->save();

}

**----- Class Book -----**

/\*\*

\* Search the book with the title and book number in the database

\* @param number book number

\* @param title title of a book

\* @return Book the book that matches the query constrains

\*/

public function getBookByTitleAndNumber(string $title, string $number)

{

return $this->where('title', 'like', '%'.$title.'%')

->where('book\_number', 'like', '%'.$number.'%')->first();

}